

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	:	Before the Examiner:
McKnight et al.	:	Patel, Manglesh M.
	:	
Serial No.: 10/773,497	:	Group Art Unit: 2178
	:	
Filing Date: February 6, 2004	:	
	:	IBM Corporation
Title: METHOD FOR	:	Dept. T81/Bldg. 503
CAPTURING DOCUMENT	:	P.O. Box 12195
STYLE BY EXAMPLE	:	3039 Cornwallis Road
	:	Research Triangle Park, NC 27709

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. **REAL PARTY IN INTEREST**

The real party in interest is International Business Machines Corporation, which is the assignee of the entire right, title and interest in the above-identified patent application.

II. **RELATED APPEALS AND INTERFERENCES**

There are no other appeals or interferences known to Appellants, Appellants' legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. **STATUS OF CLAIMS**

Claims 14-15 are pending in the Application. Claims 1-13 and 16-44 were cancelled. Claims 14-15 stand rejected. Claims 14-15 are appealed.

IV. STATUS OF AMENDMENTS

Appellants have submitted an amendment (January 30, 2008) canceling claims 37-39 and 44 following receipt of the final office action (October 30, 2007).

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 14:

In one embodiment of the present invention, a method for generating an output document in a user preferred style, the method comprising the step of reading an example file representing the user preferred style into an input buffer. Specification, page 10, lines 16-17; Specification, page 12, lines 4-5; Figure 1, elements 100, 104. The method further comprises searching the input buffer for a pattern that matches that of an expected section. Specification, page 24, lines 10-11. If the pattern is found, from the position of the pattern, defining a first bound by searching backwards in the buffer until a previous expected search pattern is found. Specification, page 24, lines 14-15. If the pattern is found, from the position of the pattern, defining a second bound by searching forwards in the buffer until a next expected search pattern is found. Specification, page 24, lines 18-19. If the pattern is found, copying a string of characters contained within the input buffer between the first bound and the second bound to a template buffer. Specification, page 24, lines 22-24. If the pattern is found, parsing the template buffer to isolate expected keywords, and names and subsections. Specification, page 24, lines 28-29. If the pattern is found, if the expected section is a section that is repeated in a document, saving in the template buffer the line offsets of keywords, names and other elements. Specification, page 24, line 30 – page 25, line 4. If the pattern is found, replacing content-specific subsections with macro names. Specification, page 25, lines 6-8. If the pattern is not found, creating a default template buffer for the expected section. Specification, page 25, lines 21-22.

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 14-15 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Lie et al. ("Cascading Style Sheets, Level 1, W3C," Jan. 1999, pgs 1-70) (hereinafter "Lie").

VII. ARGUMENT

A. Claims 14-15 are not properly rejected under 35 U.S.C. §103(a) as being unpatentable over Lie.

1. Claim 14 is patentable over Lie for at least the below-mentioned reasons.

The Examiner cites pages 6 and 15 of Lie as teaching "reading an example file representing said user preferred style into an input buffer" as recited in claim 14. Office Action (6/18/2007), pages 3-4; Office Action (10/30/2007), pages 3-4. Appellants respectfully traverse.

Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Page 6. Lie further teaches that the author's rules have a higher priority than the readers' rules. Page 15. Lie additionally teaches that any rules specified in the sheet itself override rules in imported style sheets. Page 15.

Hence, Lie teaches attaching style (e.g., fonts, colors and spacing) to HTML documents.

There is no language in the cited passages that teaches reading an example file representing the user preferred style. Neither is there any language in the cited passages that teaches reading an example file representing a user preferred style into an input buffer. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

In response to Appellants' above arguments, the Examiner asserts that Lie "teaches an example file which is the CSS defined by a user and includes all the user

preferred style which is read into an input buffer or memory via parsing operation done by the browser." Office Action (10/30/2007), page 6. Appellants would like to first briefly discuss herein what is the meaning of "CSS" which the Examiner relies upon in his rejections of claims 14 and 15. Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in a markup language. See http://en.wikipedia.org/wiki/Cascading_Style_Sheets¹. CSS most common application is to style web pages written in HTML. *Id.* CSS is used to help readers of web pages to define colors, fonts, layout, and other aspects of document presentation. *Id.* CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation (written in CSS). *Id.* This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, and reduce complexity and repetition in the structural content. *Id.* CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. *Id.* CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. *Id.*

Hence, CSS is a stylesheet language that allows the user to style a web page (e.g., define colors, fonts, layout) written in HTML. CSS does not include reading an example file representing a user preferred style into an input buffer or any of the other limitations in claims 14 and 15 as discussed further below.

Further, the Examiner appears to focus on Sections 1 and 1.1 on page 6 of Lie as support for the Examiner's assertion that CSS teaches the above-cited claim limitation. Office Action (10/30/2007), page 6. Lie instead teaches designing simple style sheets using HTML. Section 1, page 6. Lie further teaches an example of

¹ Appellants are not conceding that anything in the Wikipedia article constitutes prior art. Further, Appellants are referencing the Wikipedia article only for whatever value it has to interpret the understanding of the concept of Cascading Style Sheets as one of ordinary skill in the art reading Lie in

setting the text color of H1 elements to blue. *Id.* Lie additionally teaches that HTML authors need to write style sheets only if they want to suggest a specific style for their documents. *Id.* Furthermore, Lie teaches ways to combine style and HTML, such as by using a "LINK" element to link an external style sheet, a "STYLE" element inside the "HEAD" element; an imported style sheet using the CCS "@import" notation; and a "STYLE" attribute on an element inside "BODY." Section 1.1, page 6.

Hence, Lie teaches attaching style to HTML documents by using a "LINK" element to link an external style sheet, a "STYLE" element inside the "HEAD" element; an imported style sheet using the CCS "@import" notation; and a "STYLE" attribute on an element inside "BODY."

There is no language in the cited passages that teaches reading an example file representing the user preferred style. Instead, Lie teaches using a LINK element to link an external style sheet, where an external style sheet is a sheet of computer language used to describe the presentation of structured documents. *See* http://en.wikipedia.org/wiki/Stylesheet_language². For example, as illustrated on page 6 of Lie, "H1 {color:blue}" is an example of CSS syntax used to set the text color of H1 elements to blue. *See also Id.* A style sheet containing such CSS syntax is not the same as an example file representing the user preferred style, as discussed in Appellants' Specification. Neither is there any language in the cited passages that teaches reading an example file representing a user preferred style into an input buffer. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

As understood by Appellants, the Examiner cites 6 and 15 of Lie as teaching "searching said input buffer for a pattern that matches that of an expected section" as

¹ 1999 would have had.

² Appellants are not conceding that anything in the Wikipedia article constitutes prior art. Further, Appellants are referencing the Wikipedia article only for whatever value it has to interpret the understanding of the concept of a style sheet language as one of ordinary skill in the art reading Lie in 1999 would have had.

recited in claim 14. Office Action (6/18/2007), pages 3-4; Office Action (10/30/2007), pages 3-4. Appellants respectfully traverse.

As stated above, Lie instead teaches attaching style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that any rules specified in the sheet itself override rules in imported style sheets.

There is no language in the cited passages that teaches searching an input buffer. Neither is there any language in the cited passages that teaches searching an input buffer for a pattern that matches that of an expected section. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

The Examiner does not specifically address Appellants' contention that there is no language in Lie that teaches searching the input buffer for a pattern that matches that of an expected section. As discussed above, CSS involves allowing the user to attach style (e.g., fonts, colors and spacing) to HTML documents. CSS does not involve searching an input buffer. Neither does CSS involve searching an input buffer for a pattern that matches that of an expected section.

As understood by Appellants, the Examiner admits that Lie does not teach "if said pattern is found, from the position of said pattern, defining a first bound by searching backwards in said buffer until a previous expected search pattern is found" as recited in claim 14. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. As further understood by Appellants, the Examiner admits that Lie does not teach "from the position of said pattern, defining a second bound by searching forwards in said buffer until a next expected search pattern is found" as recited in claim 14. *Id.* As further understood by Appellants, the Examiner asserts that the above-cited claim limitations are well known in the art. *Id.* The reasoning for modifying Lie to include the above-cited missing claim limitations is "to apply a CSS to an HTML document by searching the external stylesheet selectors, hence

resulting in an output document in a user preferred style." *Id.* Appellants respectfully traverse.

Appellants respectfully traverse the assertion that the limitation of "if said pattern is found, from the position of said pattern, defining a first bound by searching backwards in said buffer until a previous expected search pattern is found" is well known in the art. Further, Appellants respectfully traverse the assertion that the limitation of "from the position of said pattern, defining a second bound by searching forwards in said buffer until a next expected search pattern is found" is well known in the art. Appellants had requested the Examiner to provide a reference that teaches that if the pattern is found, from the position of the pattern, defining a first bound by searching backwards in the buffer until a previous expected search pattern is found pursuant to M.P.E.P. §2144.03. Response (8/14/2007), page 8. However, the Examiner did not provide such a reference. Further, Appellants had requested the Examiner to provide a reference that teaches that if the pattern is found, from the position of the pattern, defining a second bound by searching forwards in the buffer until a next expected search pattern is found pursuant to M.P.E.P. §2144.03. Response (8/14/2007), pages 8-9. However, the Examiner did not provide such a reference.

Furthermore, the Examiner has not provided any rational underpinning as to how the Examiner derived his reasoning for modifying Lie to include the above-cited missing claim limitations. The Examiner simply states "to apply a CSS to an HTML document by searching the external stylesheet selectors, hence resulting in an output document in a user preferred style" as reasoning for modifying Lie to include the above-cited claim limitations. While the Examiner may consider many factors in finding a reason to combine, the Examiner still must explain how the Examiner derived the reasoning for modifying Lie to include the above-cited missing claim limitations. *KSR International Co. v. Teleflex Inc.*, 82 U.S.P.Q.2d 1385, 1396 (U.S. 2007). Consequently, the Examiner's reasoning for modifying Lie to include the

missing claim limitations of claims 14 is insufficient to support a *prima facie* case of obviousness for rejecting claims 14-15. *Id.*

Further, Lie is a publication by the World Wide Web Consortium ("W3C") to promote the deployment of the recommendation of the cascading style sheet mechanism which allows authors and readers to attach style to HTML documents. Abstract. The Examiner has not provided any reasons as to why one skilled in the art would modify Lie (which promotes the deployment of the recommendation of the cascading style sheet mechanism which allows authors and readers to attach style to HTML documents) to define a first bound by searching backwards in the buffer until a previous expected search pattern is found if the pattern is found, from the position of the pattern (missing claim limitation). Neither has the Examiner provided any reasons as to why one skilled in the art would modify Lie (which promotes the deployment of the recommendation of the cascading style sheet mechanism which allows authors and readers to attach style to HTML documents) to define a second bound by searching forwards in the buffer until a next expected search pattern is found if the pattern is found, from the position of the pattern (missing claim limitation). The Examiner's rationale ("to apply a CSS to an HTML document by searching the external stylesheet selectors, hence resulting in an output document in a user preferred style") does not provide such reasoning.

Why would the reason to modify Lie (which promotes the deployment of the recommendation of the cascading style sheet mechanism which allows authors and readers to attach style to HTML documents) to define a first bound by searching backwards in the buffer until a previous expected search pattern is found if the pattern is found, from the position of the pattern (missing claim limitation) be to search the external stylesheet selectors? Further, why would the reason to modify Lie (which promotes the deployment of the recommendation of the cascading style sheet mechanism which allows authors and readers to attach style to HTML documents) to define a second bound by searching forwards in the buffer until a next expected search pattern is found if the pattern is found, from the position of the pattern

(missing claim limitation) be to search the external stylesheet selectors? Lie is not concerned with searching the external stylesheet selectors.

Further, what is the rational connection between searching the external stylesheet selectors (Examiner's reasoning) and defining a first bound by searching backwards in the buffer until a previous expected search pattern is found if the pattern is found, from the position of the pattern (missing claim limitation)? Further, what is the rational connection between searching the external stylesheet selectors (Examiner's reasoning) and defining a second bound by searching forwards in the buffer until a next expected search pattern is found if the pattern is found, from the position of the pattern (missing claim limitation)?

Hence, the Examiner's rationale does not provide reasons that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would modify Lie to include the missing claim limitations of claims 14. Accordingly, the Examiner has not presented a *prima facie* case of obviousness for rejecting claims 14-15. *KSR International Co. v. Teleflex Inc.*, 82 U.S.P.Q.2d 1385, 1396 (U.S. 2007); *In re Rouffet*, 47 U.S.P.Q.2d 1453, 1458 (Fed. Cir. 1998).

In response to Appellants' above arguments, the Examiner appears to assert that the braces "{ }" as shown on page 6 of Lie teaches the claimed first and second bounds. Office Action (10/30/2007), page 6. Appellants respectfully traverse. As stated above, "H1 {color:blue}" on page 6 of Lie is an example of CSS syntax used to set the text color of H1 elements to blue. These "{ }" are not the claimed first and second bounds but instead are programming language syntax.

Additionally, as understood by Appellants, the Examiner cites pages 6 and 15 of Lie as teaching "copying a string of characters contained within said input buffer between said first bound and said second bound to a template buffer" as recited in claim 14. Office Action (6/18/2007), pages 3-4; Office Action (10/30/2007), pages 3-4. Appellants respectfully traverse.

As stated above, Lie instead teaches attaching style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that any rules specified in the sheet itself override rules in imported style sheets.

There is no language in the cited passages that teaches copying a string of characters. Neither is there any language in the cited passages that teaches copying a string of characters contained within the input buffer. Neither is there any language in the cited passages that teaches copying a string of characters contained within the input buffer between the first bound and the second bound. Neither is there any language in the cited passages that teaches copying a string of characters contained within the input buffer between the first bound and the second bound to a template buffer. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Furthermore, as understood by Appellants, the Examiner cites pages 6 and 15 of Lie as teaching "parsing said template buffer to isolate expected keywords, and names and subsections" as recited in claim 14. Office Action (6/18/2007), pages 3-4; Office Action (10/30/2007), pages 3-4. Appellants respectfully traverse.

As stated above, Lie instead teaches allowing authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that any rules specified in the sheet itself override rules in imported style sheets.

There is no language in the cited passages that teaches parsing a template buffer. Neither is there any language in the cited passages that teaches parsing a template buffer to isolate expected keywords and names and subsections. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Further, as understood by Appellants, the Examiner cites pages 6 and 15 of Lie as teaching "if said expected section is a section that is repeated in a document, saving in said template buffer the line offsets of keywords, names and other elements"

as recited in claim 14. Office Action (6/18/2007), pages 3-4; Office Action (10/30/2007), page 3-4. Appellants respectfully traverse.

As stated above, Lie instead teaches allowing authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that any rules specified in the sheet itself override rules in imported style sheets.

There is no language in the cited passage that teaches saving in the template buffer the line offsets of keywords, names and other elements. Neither is there any language in the cited passage that teaches saving in the template buffer the line offsets of keywords, names and other elements if the expected section is a section that is repeated in a document. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Furthermore, as understood by Appellants, the Examiner cites pages 6 and 15 of Lie as teaching "replacing content-specific subsections with macro names" as recited in claim 14. Office Action (6/18/2007), pages 3-4; Office Action (10/30/2007), pages 3-4. Appellants respectfully traverse.

As stated above, Lie instead teaches allowing authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that any rules specified in the sheet itself override rules in imported style sheets.

There is no language in the cited passages that teaches replacing content-specific subsections with macro names. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Additionally, as understood by Appellants, the Examiner cites pages 6 and 15 of Lie as teaching "if said pattern is not found, creating a default template buffer for said expected section" as recited in claim 14. Office Action (6/18/2007), pages 3-4; Office Action (10/30/2007), pages 3-4. Appellants respectfully traverse.

As stated above, Lie instead teaches allowing authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that any rules specified in the sheet itself override rules in imported style sheets.

There is no language in the cited passages that teaches creating a default template buffer. Neither is there any language in the cited passages that teaches creating a default template buffer for the expected section. Neither is there any language in the cited passages that teaches creating a default template buffer for the expected section if the pattern is not found. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

2. Claim 15 is patentable over Lie for at least the reasons that claim 14 is patentable over Lie.

Claim 15 recites combinations of features of independent claim 14, and hence claim 15 is patentable over Lie for at least the above-stated reasons that claim 14 is patentable over Lie.

3. Claim 15 is patentable over Lie for at least the below-mentioned reasons.

As understood by Appellants, the Examiner cites pages 6 and 9 of Lie as teaching "getting a said template buffer for each section to be generated in said output document" as recited in claim 15. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. Appellants respectfully traverse.

Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Page 6. Lie further teaches that by using the ID attribute as a selector, one can set style properties on a per-element basis. Page 9.

There is no language in the cited passages that teaches getting a template buffer. Neither is there any language in the cited passages that teaches getting a template buffer for each section to be generated. Neither is there any language in the cited passages that teaches getting a template buffer for each section to be generated

in the output document. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

As further understood by Appellants, the Examiner cites pages 6 and 9 of Lie as teaching "getting user content for all sections of said output document" as recited in claim 15. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. Appellants respectfully traverse.

As stated above, Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that by using the ID attribute as a selector, one can set style properties on a per-element basis.

There is no language in the cited passages that teaches getting user content. Neither is there any language in the cited passages that teaches getting user content for all sections of the output document. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Furthermore, as understood by Appellants, the Examiner cites pages 6 and 9 of Lie as teaching "creating an output buffer for storing said output document" as recited in claim 15. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. Appellants respectfully traverse.

As stated above, Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that by using the ID attribute as a selector, one can set style properties on a per-element basis.

There is no language in the cited passages that teaches creating an output buffer. Neither is there any language in the cited passages that teaches creating an output buffer for storing an output document. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Additionally, as understood by Appellants, the Examiner cites pages 6 and 9 of Lie as teaching "for each section of said output document, putting a corresponding template buffer into a temporary output buffer" as recited in claim 15. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. Appellants respectfully traverse.

As stated above, Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that by using the ID attribute as a selector, one can set style properties on a per-element basis.

There is no language in the cited passages that teaches putting a corresponding template buffer into a temporary output buffer. Neither is there any language in the cited passages that teaches putting a corresponding template buffer into a temporary output buffer for each section of the output document. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Further, as understood by Appellants, the Examiner cites pages 6 and 9 of Lie as teaching "replacing macro names in said temporary output buffer with user content information" as recited in claim 15. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. Appellants respectfully traverse.

As stated above, Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that by using the ID attribute as a selector, one can set style properties on a per-element basis.

There is no language in the cited passages that teaches replacing macro names. Neither is there any language in the cited passages that teaches replacing macro names in the temporary output buffer. Neither is there any language in the cited passages that teaches replacing macro names in the temporary output buffer

with user content information. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Further, as understood by Appellants, he Examiner cites pages 6 and 9 of Lie as teaching "if this section is expected to be repeated and the user desires alignment, using corresponding template offsets to modify said temporary output buffer for aligning keywords, names, and other sub-sections" as recited in claim 15. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. Appellants respectfully traverse.

As stated above, Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that by using the ID attribute as a selector, one can set style properties on a per-element basis.

There is no language in the cited passages that teaches using corresponding template offsets to modify a temporary output buffer. Neither is there any language in the cited passages that teaches using corresponding template offsets to modify a temporary output buffer for aligning keywords, names, and other sub-sections. Neither is there any language in the cited passages that teaches using corresponding template offsets to modify a temporary output buffer for aligning keywords, names, and other sub-sections if this section is expected to be repeated and the user desires alignment. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Furthermore, as understood by Appellants, the Examiner cites pages 6 and 9 of Lie as teaching "inserting the content of said temporary output buffer into said output buffer" as recited in claim 15. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. Appellants respectfully traverse.

As stated above, Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie

further teaches that by using the ID attribute as a selector, one can set style properties on a per-element basis.

There is no language in the cited passages that teaches inserting the content of a temporary output buffer into an output buffer. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

Additionally, as understood by Appellants, the Examiner cites pages 6 and 9 of Lie as teaching "writing said output buffer to a file" as recited in claim 15. Office Action (6/18/2007), page 4; Office Action (10/30/2007), page 4. Appellants respectfully traverse.

As stated above, Lie instead teaches a sheet mechanism that allows authors and readers to attach style (e.g., fonts, colors and spacing) to HTML documents. Lie further teaches that by using the ID attribute as a selector, one can set style properties on a per-element basis.

There is no language in the cited passages that teaches writing an output buffer to a file. Therefore, the Examiner's cited passages do not teach the above-cited claim limitation as asserted by the Examiner.

VIII. CONCLUSION

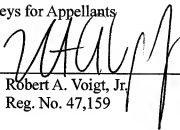
For the reasons noted above, the rejections of claims 14-15 are in error. Appellants respectfully request reversal of the rejections and allowance of claims 14-15.

Respectfully submitted,

WINSTEAD P.C.

Attorneys for Appellants

By: _____


Robert A. Voigt, Jr.
Reg. No. 47,159

P.O. Box 50784
Dallas, Texas 75201
(512) 370-2832

CLAIMS APPENDIX

14. A method for generating an output document in a user preferred style, comprising the steps of:

- reading an example file representing said user preferred style into an input buffer;

- searching said input buffer for a pattern that matches that of an expected section;

- if said pattern is found,

- from the position of said pattern, defining a first bound by searching backwards in said buffer until a previous expected search pattern is found;

- from the position of said pattern, defining a second bound by searching forwards in said buffer until a next expected search pattern is found;

- copying a string of characters contained within said input buffer between said first bound and said second bound to a template buffer;

- parsing said template buffer to isolate expected keywords, and names and subsections;

- if said expected section is a section that is repeated in a document, saving in said template buffer the line offsets of keywords, names and other elements;

- replacing content-specific subsections with macro names; and

- if said pattern is not found,

- creating a default template buffer for said expected section.

15. The method of claim 14, comprising the further steps of:

- getting a said template buffer for each section to be generated in said output document;

- getting user content for all sections of said output document;

- creating an output buffer for storing said output document;

- for each section of said output document,

- putting a corresponding template buffer into a temporary output buffer;

replacing macro names in said temporary output buffer with user content information;

if this section is expected to be repeated and the user desires alignment, using corresponding template offsets to modify said temporary output buffer for aligning keywords, names, and other sub-sections;

inserting the content of said temporary output buffer into said output buffer; and

writing said output buffer to a file.

EVIDENCE APPENDIX

No evidence was submitted pursuant to §§1.130, 1.131, or 1.132 of 37 C.F.R. or of any other evidence entered by the Examiner and relied upon by Appellants in the Appeal.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings to the current proceeding.

Austin_1 525848v.2